DirichletRank: Ranking Web Pages Against Link Spams

Xuanhui Wang[†], **Tao Tao**[†], **Jian-Tao Sun**[‡], **ChengXiang Zhai**[†]

[†]Department of Computer Science,

University of Illinois at Urbana-Champaign

{xwang20, taotao, czhai}@cs.uiuc.edu

[‡]Microsoft Research Asia, Beijing, P.R.China

jtsun@microsoft.com

Abstract

Anti-spamming has become one of the most important challenges to web search engines and attracted increasing attention in both industry and academia recently. Since most search engines now use link-based ranking algorithms, link-based spamming has become a major threaten. In this paper, we show that the popular link-based ranking algorithm PageRank, while being successfully used in the Google search engine, has a "zero-one gap" flaw, which can be potentially exploited to spam PageRank results easily. The "zero-one gap" problem arises from the current ad hoc way of computing the transition probabilities in the random surfing model. We propose a novel *DirichletRank* algorithm in a more principled way of computing these probabilities based on Bayesian estimation with a Dirichlet prior. DirichletRank is a variant of PageRank, but it does not have the problem of "zero-one gap" and is analytically shown to be substantially more resistant to link farm spams than PageRank. Simulation experiments using real web data show that, compared with the original PageRank, DirichletRank is significantly more robust against several typical link spams and is more stable under link perturbations, in general. Moreover, experiment results also show that DirichletRank

is more effective than PageRank due to its more reasonable allocation of transition probabilities. Since DirichletRank can be computed as efficiently as PageRank, it is scalable to large-scale web applications.

1 Introduction

As search engines are becoming a dominant way of information acquisition in our daily life, increased appearance of a target page in the web search results, especially on the top of the ranked lists, may yield significant financial gain for the target web site. This unavoidably leads to the emergence of web spamming [15].

Web spamming is a method to maliciously induce bias to search engines so that certain target pages will be ranked much higher than they deserve. Consequently, it leads to deteriorated quality of search results and would significantly reduce the utility of a web search engine for users. For example, when a user submits a query "Kaiser pharmacy" to a search engine to find information about pharmacies affiliated with Kaiser-Permanente, the top result may be spammed as `techdictionary.com` [15].

Indeed, anti-spamming is now a major challenge for all search engines [18, 16, 5, 25, 29, 13, 15, 1]. The early stage of Web spamming focused on page contents and achieved spamming by adding a wide variety of query keywords regardless of their relevance. This type of spamming is relatively easy to detect [13], hence has not made a significant impact on search engines. However, the use of link-based algorithms, such as PageRank [8, 23], in most search engines has stimulated the development of another type of spamming — *link spamming* [14]: spammers intentionally set up link structures, involving a lot of interconnected pages, to boost the PageRank scores of a small number of target pages. Unlike content-based spamming, not only can link spamming render much more significant ranking gains, but it is also much harder to detect on the search engine side [14, 29]. Figure 1 illustrates one of such typical link spam structures. We use *leakage* to denote the PageRank scores that reach the link farm from external pages. In this structure, a web owner creates a large number of

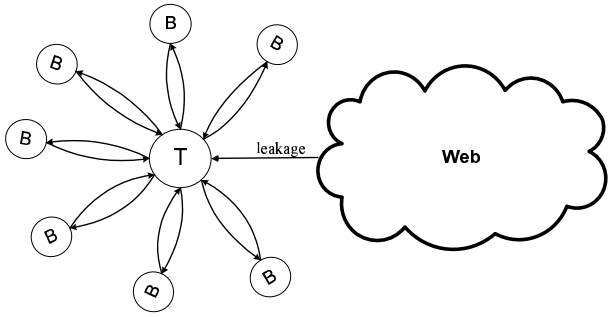


Figure 1: An example of the link-spam using bogus pages

bogus web pages B's (*i.e.*, pages whose sole purpose is to promote the target page's ranking score), all pointing to and pointed by a single target page T. The PageRank algorithm is sensitive to this type of structures [14], and tends to assign a much higher ranking score to T than it deserves (up to 10 times of the original score easily).

Despite the importance of dealing with link spamming, only a few studies have been done so far [16, 14, 25, 29, 5], mostly based on PageRank. In this paper, we show that PageRank has a "zero-one gap" flaw, which can be potentially exploited by a spammer to easily spam PageRank results. To the best of our knowledge, this flaw has not been addressed in any existing work.

The "zero-one gap" problem refers to the unreasonable dramatic difference between a page with no out-link and one with a single out-link in their probabilities of randomly jumping to any page. The problem arises from the ad hoc way of computing the transition probabilities in the random surfing model adopted in the existing work. To address this problem, we propose a novel *DirichletRank* algorithm in a more principled way of computing the transition probabilities based on Bayesian estimation with a Dirichlet prior. DirichletRank is a variant of PageRank, but it does not have the problem of "zero-one gap" and can be analytically shown to be substantially more resistant to link farm spams than PageRank. Simulation experiments using real web data show that, compared with the original PageRank, DirichletRank is significantly more robust against several typical link spams and is more stable under link perturbations, in general. Moreover, experiment results also show that Dirichle-

tRank is more effective than PageRank due to its more reasonable allocation of transition probabilities. Since DirichletRank can be computed as efficiently as PageRank, it is scalable to large-scale web applications.

The contribution of this paper includes:

1. We identify the "zero-one gap" flaw of the PageRank algorithm and discuss its impact on link spamming.

2. We derive a novel ranking algorithm DirichletRank to solve the "zero-one gap" problem based on Bayesian estimation of transition probabilities.

3. We make theoretical comparison between DirichletRank and PageRank, and prove that DirichletRank has stronger resistance against the link farm spams.

4. We do experiments to show that DirichletRank can not only combat link farms but also improve the search performance over PageRank.

The rest of the paper is organized as follows. We first discuss related work in Section 2. We then review the PageRank algorithm and discuss the problem of "zero-one gap" in Section 3. In Section 4, we present the proposed DirichletRank algorithm and analytically show its robustness in dealing with link spamming. We present our experiment results in Section 5 and conclude in Section 6. All proofs are given in the appendix.

2 Related Work

PageRank [8] and HITS [19] are two earliest link analysis algorithms using eigenvectors to identify "authoritative" pages via hyperlink information. Since then, several methods have been developed to improve the algorithm accuracy [6, 17, 20, 10]. PHITS [10] proposes statistical hubs and authorities. BHITS [6] alleviates the dominance of certain special link structures in the HITS algorithm. [20] solves the "small-in-large-out" problem of HITS, and [17] calculates topic-sensitive PageRanks based on the web page categories. However, none

of these modifications can deal with link farm spams [25]. In general, RageRank approach is shown to be relatively more stable than HITS [22], however it still suffers from several special link structures and can be easily spammed via link alliances [14].

The taxonomy of web spamming is clearly defined in [15]. Link alliance structures, in which a group of spammers collaborate to build link farms, are studied in [14] theoretically. However, detection of web spamming, especially link spamming, is a very challenging problem. Most previous anti-spamming work addresses it heuristically. For example, [13] analyzes the the statistical distribution of the web pages and treats the outliers as spam pages. [29] observes that the influence of spam structures is sensitive to the random jumping probability λ and hence proposes two heuristics of personalizing λ for spam detection. [5] assumes that only honest pages have their in-link neighbors' PageRank scores obey the power law. This method is hard to be applied when a page does not have sufficiently large number of in-link neighbors. Moreover, it is computationally expensive, and cannot be effectively used in large data sets. [25] identifies a link farm seed set through a page's out-link and in-link domains, and then makes a second expansion to identify more spams. It is very sensitive to the parameter settings.

"TrustRank" [16] and "BadRank" [2] are two PageRank-like methods to quantify pages' honesties. "TrustRank" relies on the fact that good pages seldom point to bad ones, and propagates a trust score through hyperlinks. Its major shortage is the need for human experts to identify certain good seeds. On the other side, BadRank assumes that bad pages always point to pages with high BadRank values. Clearly, this assumption may not be true since one can create a bad page which points only to authoritative pages.

Our method differs from all these heuristics in that it addresses a fundamental flaw of PageRank and achieves anti-spamming through a more robust and reasonable way of using link information. Since most existing methods are based on PageRank, our method can potentially be combined with them to achieve more effective anti-spamming.

3 The "Zero-One Gap" Flaw in PageRank

In this section, we first briefly review the PageRank algorithm, and then analyze its "zero-one gap" flaw and its vulnerability to link farm spamming.

3.1 Basic PageRank

The basic PageRank algorithm [23, 8] models the whole web as a directed graph $G(V, E)$ with a vertex set V of N pages and a directed edge set E. By collapsing multiple links between the same pair of pages, we represent this graph by an $N \times N$ binary-value adjacency matrix:

$$A = [a_{ij}]_{N \times N} \text{ where } a_{ij} = \begin{cases} 1 & \text{if } (i,j) \in E \\ 0 & \text{otherwise} \end{cases}$$

When a page u links to another page v, it implies that u's creator confers v's importance. Moreover, an important page may support its linked neighbors more than an unimportant one. Based on the above two motivations, PageRank obtains any page's importance through its in-link pages' importance iteratively.

Formally, for each page v, let N_v be the out-links of v, B_v be the in-links of v, and $r(v)$ be the PageRank score of v, respectively. We have

$$r(v) = \sum_{u \in B_v} \frac{r(u)}{|N_u|},$$

Assume M is the row normalized matrix of A, the updating of PageRank scores can be expressed as:

$$\mathbf{r} = M^T \mathbf{r}$$

When there is at least one non-zero entry in each row, M is a stochastic Markov transition matrix. Mathematically the iterative updating will converge to M's principal eigenvector. However, the convergence above is guaranteed only if M is irreducible and aperiodic [21]. In web applications, the latter is guaranteed but the former is not and may result in the "rank sink" problem. PageRank therefore introduces a uniform matrix U ($U_{ij} = \frac{1}{N}$) and

interpolates it with the original matrix M with a damping factor $1-\lambda$:

$$\tilde{M} = (1-\lambda) \cdot M + \lambda \cdot U \tag{1}$$

where λ ($0 \leq \lambda \leq 1$) is the *random jumping* probability. The improved PageRank scores are calculated as:

$$\begin{aligned} \mathbf{r} &= \tilde{M}^T \mathbf{r} \\ &= (1-\lambda) M^T \mathbf{r} + \frac{\lambda}{N} \mathbf{e}_N \end{aligned}$$

where $\mathbf{e}_N = (1, ..., 1)^T$ is a column vector consisting of N elements of 1, and λ is typically set to 0.15 [8]. The intuition behind the interpolation can be explained by a random surfing model. A surfer follows the out-links of a page with probability $1-\lambda$ and uniformly jumps to random pages with probability λ. A page's PageRank score can be interpreted as the probability that a surfer would finally visit this page after surfing the whole web for sufficiently long time.

3.2 Solving the "Zero Out-Links" Problem

The basic PageRank assumes each row of the matrix M has at least one nonzero entry, *i.e.* the corresponding vertex in G has at least one out-link. Unfortunately, this assumption never holds in reality. Many web pages simply have no out-link at all. Moreover, many web applications only consider a subgraph of the whole web. In these cases, even if a page has out-links, these out-links may have been removed when the whole web is projected to a subgraph. For example, in the WT10G[1] data, only $1,295,841$ out of $1,692,096$, roughly 77%, documents have out-links. Simply removing all the pages without out-links is not a solution because it generates new zero-out-link pages. Indeed, this "dangling page" problem has been identified in [12, 7, 8, 23]. [7] analyzes previous solutions and shows that they all boil down to the following approach:

$$\tilde{M} = (1-\lambda) \cdot M + \tilde{\lambda} \cdot U \tag{2}$$

[1] http://es.csiro.au/TRECWeb/wt10g.html

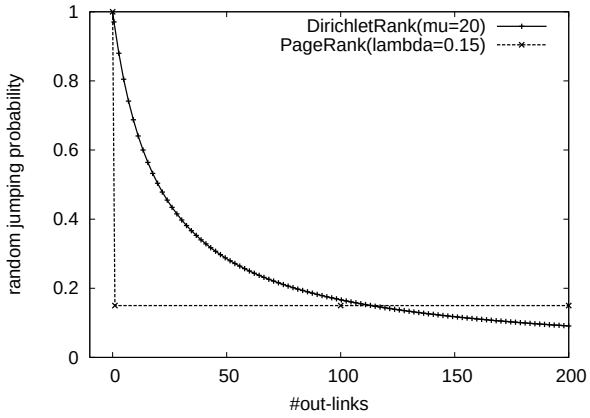


Figure 2: The random jumping probability w.r.t the number of out-links

where

$$\tilde{\lambda}(i) = \begin{cases} \lambda & \text{if } \sum_j M_{ij} = 1, \\ 1 & \text{otherwise.} \end{cases}$$

$\tilde{M}$ is a Markov matrix and guarantees the equilibrium distribution [12]. PageRank scores are:

$$\mathbf{r} = (1-\lambda)M^T\mathbf{r} + \frac{\tau}{N}\mathbf{e}_N$$

where τ is the weighted sum of the random jumping probabilities,

$$\tau = \sum_{i=1}^{N} r(i) \times \tilde{\lambda}(i)$$

when $\tilde{\lambda}(i) = \lambda$ for $1 \leq i \leq N$, $\tau = \lambda$.

As proved in [7], Equations (1) and (2) are equivalent in terms of final page ranking results, even if $\tilde{M}$ in 1 is not a Markov matrix. In the rest of the paper, we use Equation (2) as the formula for PageRank.

3.3 The "Zero-One Gap" Problem

Although Equation (2) solved the zero-out-link problem, there is another problem to be addressed: The probability of jumping to random pages is 1 in a zero-out-link page, but it drops to λ (in most cases, $\lambda = 0.15$) for a page with a single out-link. We illustrate this

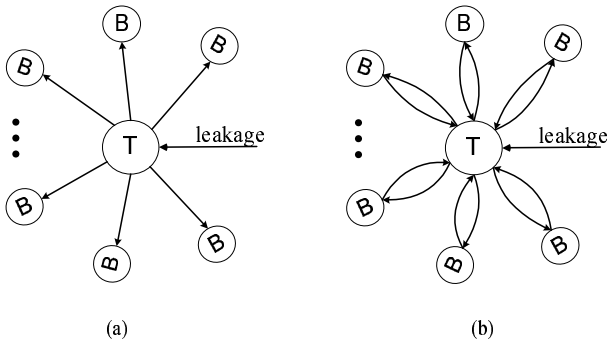


Figure 3: Two contrast structures

problem in Figure 2. The dashed line in Figure 2 illustrates the random jumping probability of the PageRank algorithm with respect to the number of out-links. Clearly there is a big "gap" between 0 and 1 out-link. We refer to this problem as "zero-one gap".

The zero-one gap problem represents a serious flaw in PageRank as it allows a spammer to easily manipulate PageRank to achieve spamming. Consider the structures plotted in Figure 3, which illustrates the spamming process. While (a) is the case without link spamming, (b) represents a typical spamming structure with all bogus pages B's having back links to the target page T. We denote $r_o(\cdot)$ as the PageRank score in Figure 3 (a). In contrast, $r_s(\cdot)$ denotes the PageRank score in Figure 3 (b). We then simply extend the proof in [14] to the following theorem.

Theorem 1 *With k bogus pages, σ leakage, and τ as the weighted sum of the random jumping probabilities,*

$$r_o(T) = \sigma + \frac{\tau}{N} \tag{3}$$

$$r_s(T) = \frac{1}{2\lambda - \lambda^2}\left[\sigma + \frac{\tau(k(1-\lambda)+1)}{N}\right], \tag{4}$$

and $r_s(T) \geq \frac{1}{2\lambda-\lambda^2} r_o(T)$ for any positive integer k.

The proof of Theorem 1 is given in the appendix and so do all following theorems.

Theorem 1 shows that $r_s(T)$ is constantly larger than or equal to $r_o(T)$ for any positive integer k: Given $\lambda \in [0,1]$, $\frac{\partial}{\partial\lambda}\frac{1}{2\lambda-\lambda^2} = \frac{-2(1-\lambda)}{(2\lambda-\lambda^2)^2} \leq 0$. Thus, when $\lambda = 1$, $\frac{1}{2\lambda-\lambda^2}$ reaches its

minimum value 1. We conclude $r_s(T) \geq r_o(T)$ over the range of all λ values. On the other hand, $\lim_{\lambda \to 0} \frac{1}{2\lambda - \lambda^2} = \infty$. This means a small λ, usually preferred in PageRank [23], can result in that $r_s(T)$ is much larger than $r_o(T)$. For example, $\lambda = 0.15$ makes $r_s(T)$ be about 3 times larger than $r_o(T)$. Note that the above discussion applies for all k's, even $k = 1$. When $k > 1$, the difference will be even aggravated.

The discussion above clearly indicates an intrinsic flaw in PageRank: when $k = 1$, there is only one bogus page in Figure 3 (b), but the addition of this bogus page makes the PageRank score of the target page 3 times larger than before. This is totally because a surfer is forced to jump back to the target page with a high probability in Figure 3 (b). Indeed, given a default value $\lambda = 0.15$, the single out-link in a bogus page forces a surfer to jump back to the target page with a probability 0.85!

The analysis above demonstrates that the "zero-one gap" problem represents a serious flaw of PageRank, which makes it sensitive to a local structure change and thus vulnerable to link spamming.

4 DirichletRank Algorithm

In this section, we derive a new algorithm called DirichletRank based on Bayesian estimation of transition probabilities. DirichletRank not only solves the problem of zero-one gap, but also provides a more principled way to solve the original zero-out-link problem. We analytically compare DirichletRank with PageRank and show that DirichletRank is less sensitive to local structure changes and more robust than PageRank.

4.1 Bayesian Estimation of Transition Probabilities

We note that the zero-one gap problem is caused by an unreasonable allocation of probabilities in the random surfing model. A natural solution would be to seek for a more principled way to set such probabilities.

Let us assume that, in our random surfing model, the probabilities of a surfer transiting from a page v to other pages follow a multinomial distribution Θ_v. We may treat all v's out-links L_v as a sample of this hidden distribution. A maximum likelihood estimator can then be used to estimate the Θ_v, giving us precisely the M discussed in the previous section. However, the maximum likelihood estimator generates many undesirable zero probabilities due to the small size of the sample. To solve this problem, we put a prior on Θ_v and use the Bayesian estimator. Specifically, we define a Dirichlet prior distribution on Θ_v with hyperparameter $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_N)$, given by

$$Dir(\Theta_v|\alpha) = C(\alpha) \prod_{i=1}^{N} \theta_i^{\alpha_i - 1}$$

where $C(\alpha)$ is the normalization factor. The parameters α_i are chosen to be $\alpha_i = \mu \cdot P_{rand}$ where μ is a parameter and $P_{rand} = \frac{1}{N}$ is the uniform jumping probability. Since Dirichlet is a conjugate prior for multinomial distribution, the Bayesian estimate of the probability that a surfer will select link l after page v is

$$P(l|L_v) = \int P(l|\Theta_v) P(\Theta_v|L_v) d\Theta_v$$

where the posterior probability is given by

$$P(\Theta_v|L_v) \propto \prod_{\text{all links } l} P(l|\Theta_v)^{c(l,L_v)+\mu P_{rand}-1}$$

which is also Dirichlet, with parameters $\tilde{\alpha}_i = c(l, L_v) + \mu P_{rand}$ where $c(l, L_v)$ is the number of times link l appears in L_v. Using the fact that the Dirichlet mean is $\frac{\tilde{\alpha}_i}{\sum_k \tilde{\alpha}_k}$, we will have:

$$\begin{aligned} P(l|L_v) &= \frac{c(l, L_v) + \mu P_{rand}}{|L_v| + \mu} \\ &= (1 - \frac{\mu}{|L_v| + \mu}) \frac{c(l, L_v)}{|L_v|} + \frac{\mu}{|L_v| + \mu} P_{rand} \\ &= (1 - \omega_v) P_{ml} + \omega_v P_{rand} \end{aligned}$$

where P_{ml} is the maximum likelihood estimator and $\omega_v = \frac{\mu}{|L_v| + \mu}$. Note that $|L_v|$ equals to the sum of the elements of the row corresponding to page v in A. In a Markov transition matrix form, we have

$$\tilde{M} = diag\{1 - \omega_1, ..., 1 - \omega_N\} \cdot M + diag\{\omega_1, ..., \omega_N\} \cdot U$$

The ranking scores can be calculated by solving the eigenvector equation:

$$\mathbf{r} = \tilde{M}^T \mathbf{r}$$

The i-th value in vector $\mathbf{r}$ is the ranking score of the i-th web page. Since we use Dirichlet prior to calculate the ranking values, we call our algorithm "*DirichletRank*". From the definition of ω_v, we can see that the larger the $|L_v|$ is, the larger $1 - \omega_v$ will be. Thus in DirichletRank, a surfer would more likely follow the out-links of the current page if the page has many out-links.

The derivation above is analogous to a similar derivation of the Dirichlet prior smoothing method in information retrieval [27, 28], which has also been shown to be quite effective for retrieval.

4.2 Comparison with PageRank

Bayesian estimation provides a principled way for setting the transition probabilities and we now show that it not only solves the zero out-link problem in an elegant way, but also solves the problem of "zero-one gap" naturally.

The random jumping probability of DirichletRank is

$$\omega(n) = \frac{\mu}{n + \mu}, 0 \leq n \leq \infty$$

where n is number of out-links and μ is the Dirichlet prior parameter. We set $\mu = 20$ and plot $\omega(n)$ in Figure 2. The figure shows that the jumping probability in DirichletRank is smoothed and no gap between 0 and 1 out-link.

To compare with the PageRank algorithm, we define $d_o(\cdot)$ and $d_s(\cdot)$ the DirichletRank scores for the structures in Figure 3 (a) and (b) respectively. We then have the corresponding theorem.

Theorem 2 *With k bogus pages, σ leakage, and τ as the weighted sum of the random jump-*

ing probabilities,

$$d_o(T) = \sigma + \frac{\tau}{N} \quad (5)$$

$$d_s(T) = \left[1 + \frac{k}{\mu^2 + (k+1)\mu}\right]\left[\sigma + \frac{k+\mu+1}{\mu+1}\frac{\tau}{N}\right] \quad (6)$$

and $d_s(T) \geq d_o(T)$ *for any positive integer* k*.*

On the surface, we obtain the similar conclusion to the PageRank scores: $d_s(T)$ is constantly larger than or equal to $d_o(T)$. However, $d_s(T)$ is in fact very close to $d_o(T)$. For example, we set $\mu = 20$, $k = 1$,

$$d_s(T) \approx \left[1 + \frac{1}{20^2 + 2 \times 20}\right] d_o(T) \approx d_o(T).$$

This indicates no significant change in T's DirichletRank scores before and after spamming. Thus DirichletRank is indeed more stable and less sensitive to the change of local structure.

We further analyze the influence of k on both PageRank and DirichletRank. In PageRank,

$$\begin{aligned} r_s(T) &= \frac{1}{2\lambda - \lambda^2}\left[\sigma + \frac{\tau(k(1-\lambda)+1)}{N}\right] \\ &= \left[\frac{1-\lambda}{2\lambda-\lambda^2}\frac{\tau}{N}\right]k + \frac{1-\lambda}{2\lambda-\lambda^2}\left[\sigma + \frac{\tau}{N}\right] \end{aligned}$$

In DirichletRank, $1 + \frac{k}{\mu^2+(k+1)\mu} < 1 + \frac{1}{\mu}$, thus,

$$\begin{aligned} d_s(T) &= \left[1 + \frac{k}{\mu^2 + (k+1)\mu}\right]\left[\sigma + \frac{k+\mu+1}{\mu+1}\frac{\tau}{N}\right] \\ &< \left[1 + \frac{1}{\mu}\right]\left[\sigma + \frac{k+\mu+1}{\mu+1}\frac{\tau}{N}\right] \\ &= \left[\frac{1}{\mu}\frac{\tau}{N}\right]k + \frac{\mu+1}{\mu}\left[\sigma + \frac{\tau}{N}\right] \end{aligned}$$

In PageRank, the scores of target pages increase linearly with the number of bogus pages k and the coefficient is $c_r = \frac{1-\lambda}{2\lambda-\lambda^2}\frac{\tau}{N}$. In DirichletRank, the scores of target pages are upper-bounded by a linear function with the coefficient $c_d = \frac{1}{\mu}\frac{\tau}{N}$. A typical setting $\lambda = 0.15$ leads to $c_r = 3.06\frac{\tau}{N}$. On the other hand, even we set μ as the smallest value 1, $c_d \leq 1 \times \frac{\tau}{N}$ is increasing much more slowly. In fact, a typical μ, as our experiment show later, is 20, which

leads to $c_d \leq 0.05\frac{\tau}{N}$. $0.05 \ll 3.06$. Thus, if a spammer creates the same number of bogus pages, the influence on DirichletRank is much less than that on PageRank.

It is interesting to note that in DirichletRank a surfer will follow the out-links of a page with high probability if the page has a large number of out-links. Intuitively, this is also reasonable, since a page with more out-links is presumably a good hub page for directing surfers to good authority pages [19], and thus it is natural to believe that the surfer will follow the out-links of such pages with a higher probability [24].

One important feature of DirichletRank is that it introduces no extra time cost. This is indeed an advantage over some other heuristics for anti-spamming, which are often computationally expensive, and it makes DirichletRank suitable for the web-scale applications.

4.3 Effect on Anti-Link-Spamming

In this section, we use a typical link spam structure plotted in Figure 3 (b) to show that DirichletRank is more resistant to link spamming than PageRank.

As discussed in the previous section, a surfer in DirichletRank randomly jumps away in a higher probability if the page has fewer out-links. Therefore, a single reversed link from each B to T fails to entrap the surfer in the local spam structure. Naturally, a spammer may want to break the DirichletRank algorithm by setting more out-links in each bogus page, but making them point to only other bogus pages. On the surface, this structure reduces randomly jumping probabilities, and hence keep the probability mass in the local structure. However, the following theorem shows this intention does not work at all.

Theorem 3 *DirichletRank $d(T)$ is independent of the internal connections between bogus pages (B's) when the following three conditions hold:*

1) Target page T has a link to each of its bogus page B;

2) Each B has a reversed link back to T;

3) Both T and B's have no out-links to pages except T and B's.

This theorem indicates that complex local structures may keep probability mass within

a local structure, but it is not able to leverage the score of target page T. In the next theorem, we show the above situation is already the best that a spammer can do. We make the assumption that no leakage goes to bogus pages B's. This assumption is reasonable since bogus pages are created intentionally for boosting a target page.

Theorem 4 *Given a fixed number of bogus pages k, when no leakage is added into any bogus page, the optimal DirichletRank score for any spamming structure is*

$$d_s(T) = \left[1 + \frac{k}{\mu^2 + (k+1)\mu}\right]\left[\sigma + \frac{k+\mu+1}{\mu+1}\frac{\tau}{N}\right]. \tag{7}$$

According to Theorem 3, adding any link between B's is an equivalently optimal spamming structure. Theorem 4 further claims that any additional out-links from bogus pages to the outside global web can only make a structure sub-optimal. Thus, the best that a spammer can do is to set up a spamming structure such as Figure 3 (b). However, in the previous subsection, we have demonstrated that DirichletRank score of a target page with such an optimal spamming structure is close to the score without any spamming.

Increasing the number of bogus pages is one way to increase the targets' DirichletRank scores. But as we analyzed before, the coefficient of the number of bogus pages in DirichletRank is much lower, indicating a spammer needs to set up much more bogus pages. More bogus pages cost the spammer more effort, and also make spam structures much easier to detect. In experiment section, we will empirically study the impact of number of bogus pages.

Above we theoretically demonstrate that DirichletRank is more resistant to several typical spam structures. In general, it is not very possible to enumerate all such structures. But till now, most link spam structures and anti-link-spamming techniques studied in previous works are on the basis of PageRank, and they are affected by the "zero-one gap" problem. DirichletRank solves the "zero-one gap" problem, thus can replace PageRank and provide a more sound basis. For example, the "collision" structure studied in [29] can not entrap DirichletRank because the number of out-links in each target page is only 1 so that a surfer will jump away to a random page with a probability approaching to 1. We will demonstrate

this in our experiment section.

5 Experiment Results

In this section, we use the TREC ".GOV" data set [3] to compare DirichletRank with PageRank empirically in terms of their robustness against link spamming and their accuracy in ranking web pages. The TREC ".GOV" data set is about 18 Gigabytes and contains $1,247,753$ web pages crawled from the ".gov" domain. Each page has 10.46 out-links on average. $1,053,372$ documents are in html format; all the others are of non-html format (*e.g.*, pure text, pdf, postscript, Microsoft Word), thus have no out-links at all. In all the experiments, unless otherwise stated, we set $\lambda = 0.15$ for PageRank as suggested in [8] and set $\mu = 20$.

5.1 Results on Bogus-Page-Based Spams

We study the impact of bogus pages on DirichletRank and PageRank by simulating bogus-page-based spams on the .GOV data set in the following way: We first calculate the baseline spam-free ranking scores using the original clean .GOV data. We then select ten pages (*i.e.*, the $10,000$th, $20,000$th, ..., and $100,000$th) from the spam-free ranking list as our target pages. We remove the out-links of all these target pages, and for each target page, create k bogus pages, each with an in-link from and an out-link to the corresponding target page. After spamming these target pages, we re-calculate the ranking scores for all the pages on this spammed .GOV data set. We evaluate the vulnerability of DirichletRank and PageRank by comparing the changes in the rank of each target page before and after spamming and by computing the amplification factor, which is defined as the ratio of the new ranking score to the old ranking score [29].

We set $k = 10$ and plot two curves in Figure 4 to compare the rank changes of DirichletRank and PageRank. To facilitate comparison, we also plot the straight diagonal line,

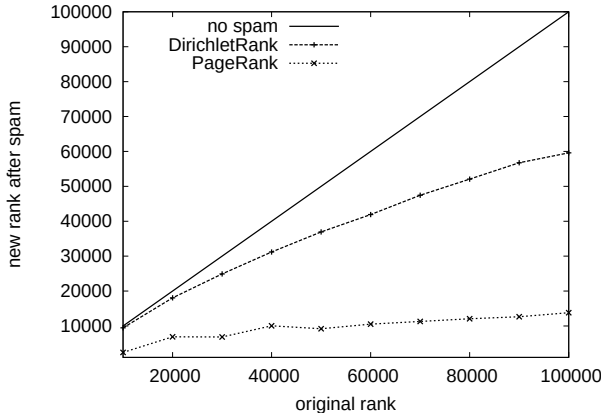
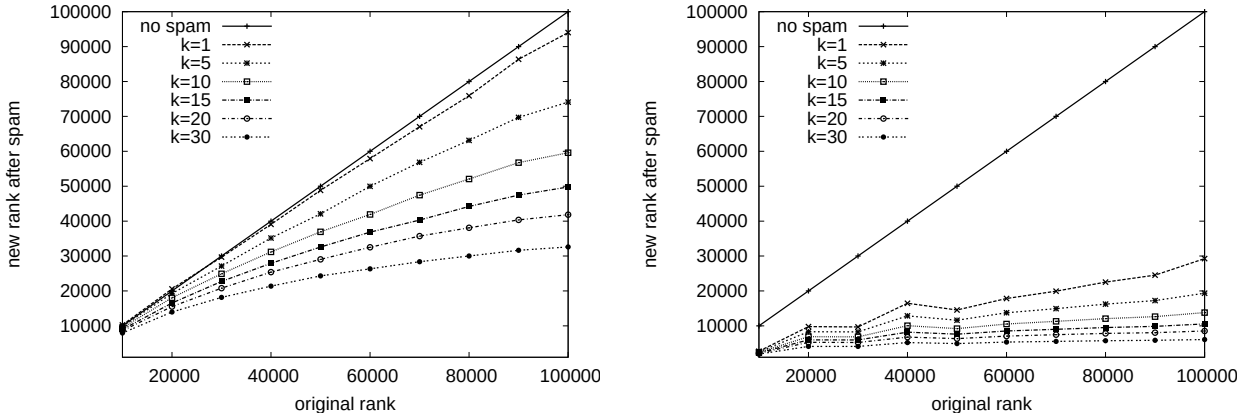


Figure 4: Comparison on the bogus page based spam



(a) DirichletRank (b) PageRank

Figure 5: Impact of number of bogus pages k

which represents the ideal case when no rank has changed; clearly, the closer a curve is to the diagonal line, the less sensitive the corresponding ranking algorithm is. We observe in Figure 4 that the DirichletRank curve is much closer to the diagonal line. For example, after spamming, the $50,000$th page is promoted to the $9,201$th by PageRank, but only to the $36,940$th by DirichletRank. We therefore conclude that DirichletRank is much less sensitive to link farm spamming.

We also study the impact of the number of bogus pages k by varying it from 1 to 30. The results are shown in Figure 5, where (a) shows ranking changes in DirichletRank and

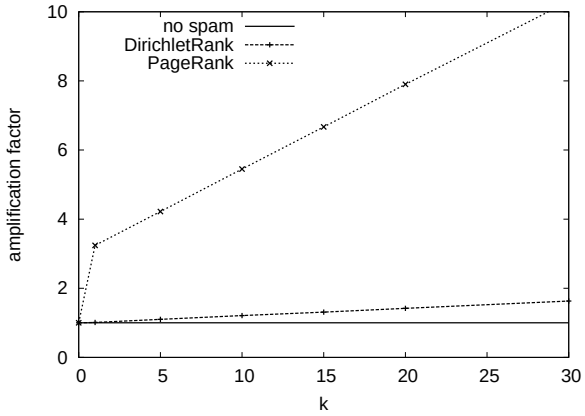


Figure 6: The amplification factor along with k

(b) shows the changes in PageRank. It is expected that a larger k would promote a target page more than a smaller k, but we can also clearly see that curves in Figure 5(a) are much closer to the diagonal line, indicating that DirichletRank is significantly less sensitive than PageRank for all the k values plotted. Indeed, the impact of 30 bogus pages on DirichletRank is still much less than that of a single bogus page on PageRank.

Figure 6 shows the average amplification factor of the ten target pages along with k. In both algorithms, the amplification factors increase roughly linearly with k, but DirichletRank has a nearly flat slope and significantly lower amplification factor values. Note that there is a jump between $k = 0$ and $k = 1$ in the PageRank curve, precisely because of the "zero-one gap" problem that we discussed at the beginning of this paper.

5.2 Results on Collusion Spams

We now study the impact of the link alliance spam structures, in which a group of spammers collaborate to build link farms [14]. In particular, we study the collusion structure identified in [29], in which a set of nodes modifying their out-links to improve each other's PageRank scores. In [29], collusion structures are detected by predefined rules, which is very time consuming yet not very accurate. We find that our DirichletRank algorithm can solve the collusion problem well without any extra computational cost.

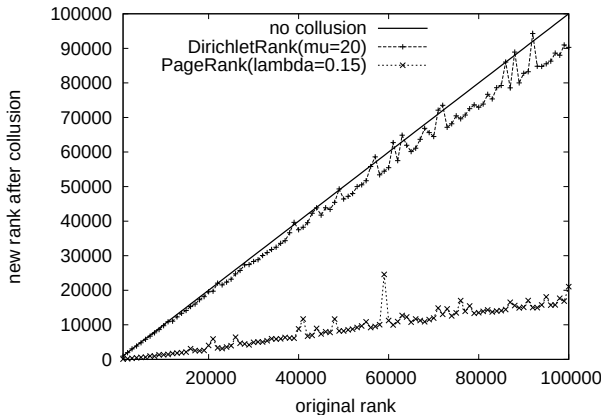


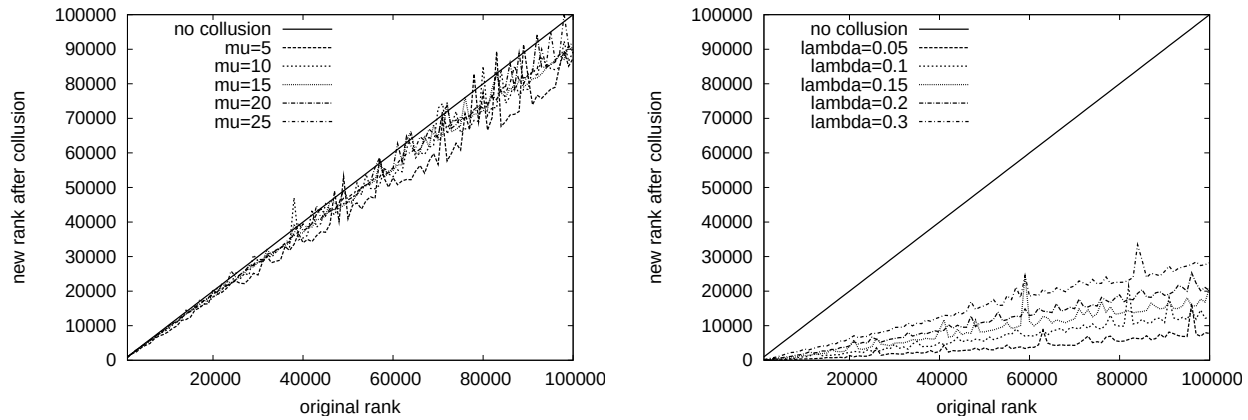
Figure 7: Comparison after creating the collusions

Similar to [29], we select 100 ranking positions, 1000th, 2000th, ..., and 100000th. At each position, we select two adjacent pages, delete all their out-links, and add two links between them with each pointing to the other. We calculate the rankings before and after the modification.

Figure 7 shows the impact of the simulated collusions. Once again, we see clearly that such collusions can not change the ranking of Dirichlet algorithm too much, but can dramatically change the PageRank ranking. Figure 8 and Figure 9 further show the impact of different values of λ and μ. We again observe that PageRank is very sensitive to collusion structures, while DirichletRank is much more stable. For example, the amplification factor in PageRank is up to 20 when $\lambda = 0.05$, but all the amplification factor values are close to 1 in DirichletRank.

5.3 Stability under Perturbation

Stability is an important property for a reliable ranking algorithm. In general, A stable ranking algorithm does not change its ranking dramatically when a small perturbation(*e.g.*, removing a small number of links or pages) is imposed [22]. In this section, we compare DirichletRank with PageRank in terms of stability. We simulate perturbation by varying the density of the links in the web graph in a way similar to how it is done in [26]. Specifically,



(a) DirichletRank (b) PageRank

Figure 8: Parameter setting for ranking changes under collusions

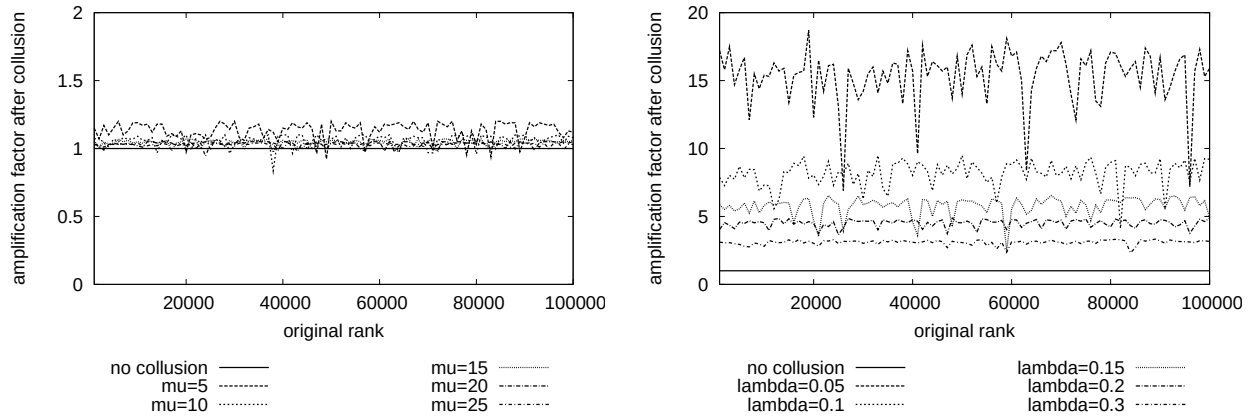
we first perturb the web graph by randomly deleting $f\%$ links with f varied from 10 to 70, and then compute PageRank and DirichletRank scores in these perturbed web graphs and compare the new ranking scores with the original ones.

Since PageRank/DirichletRank score vectors are the stationary probability distribution of a Markov matrix, we measure the difference between the two score distributions by KL-divergence [11]. Given two probability distributions $p(x)$ and $q(x)$, KL-divergence is defined as:

$$D(p||q) = \sum_x p(x) \log \frac{p(x)}{q(x)}$$

$D(p||q) \geq 0$ and $D(p||q) = 0$ iff $p = q$. In our experiment, $p(x)$ is the scores from the original graph and $q_f(x)$ is the scores from the perturbed graph.

We compare the stability of DirichletRank and PageRank in Figure 10, where the x-axis denotes the percentage of deleted links $f\%$ and y-axis denotes the KL-divergence values of $D(p||q_f)$; the smaller the divergence is, the more stable the algorithm is. We yet again observe that the divergence values of the two PageRank curves are much higher than the DirichletRank ones. This means DirichletRank curves increase much more slowly and is more stable under perturbation.



(a) DirichletRank (b) PageRank

Figure 9: Parameter setting for amplification factor under collusions

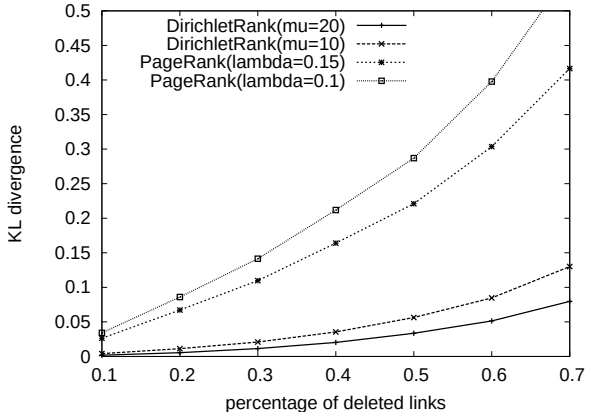


Figure 10: The stability under perturbation

5.4 Effectiveness for Web Search

The discussion above has all focused on the effectiveness of DirichletRank on dealing with link spamming. We now compare DirichletRank with PageRank in terms of their ranking accuracy. As will be shown, we find that the proposed DirichletRank algorithm is not only more resistant against link spamming, but also more accurate in ranking web pages.

To evaluate their ranking accuracy, we use the fifty "topic distillation" topics created by NIST for TREC-2003 task [4]. On average, there are 10.32 relevant documents per topic.

Table 1: Results of different ranking algorithms

Methods	AvgPrec	P@10
Text-based	0.106	0.088
PageRank (impr.)	0.134(26.4%)	0.11(25%)
DirichletRank (impr.)	0.140(31.5%)	0.116(31.8%)

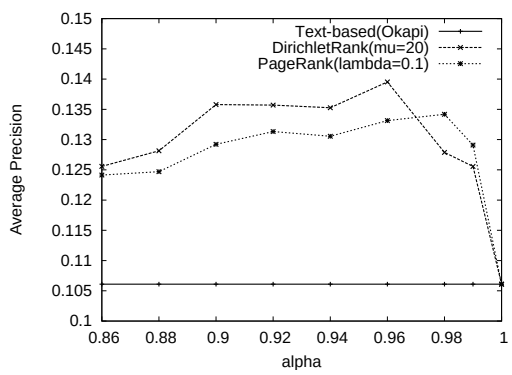
We use both non-interpolated average precision (AvgPrec) and precision at 10 documents (P@10) as our evaluation metrics. We use the Okapi retrieval method and the BM2500 weighting function for initial retrieval; the parameters are set to the same values as in [9] (*i.e.* $k1 = 4.2, k3 = 1000, b = 0.8$). We choose the top 2000 documents according to Okapi scores and construct two rankings of the documents: the ranking based on the Okapi scores and the link-based ranking. The final ranking is a combination of these two rankings [9]:

$$\alpha \cdot rank_{text} + (1 - \alpha) \cdot rank_{link}$$

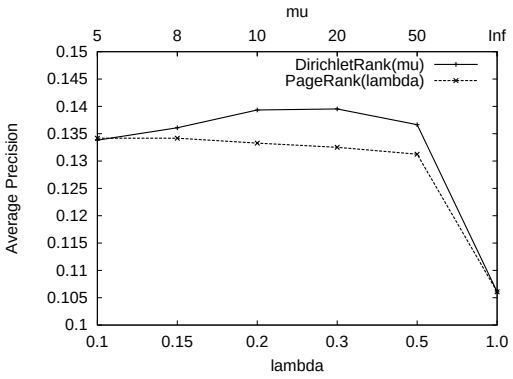
We then re-rank the 2000 documents and select the top 1000 documents for evaluation. We vary α to select the best results for both DirichletRank and PageRank.

Table 1 lists the best results of PageRank and DirichletRank. We see that both link-based ranking algorithms improve the performance significantly over the content-based method. Furthermore, DirichletRank achieves better performance than PageRank on both AvgPrec (4.03% improvement) and P@10 (5.55% improvement). A Wilcoxon signed rank test indicates the improvement on AvgPrec is statistically significant (p-value=0.034). This demonstrates that differentiating the pages with different number of out-links emphasizes the role of good hubs thus models the surfer's behavior in a better way. Note that our baseline result is not the same as the baseline in [9] because we use all the documents in ".GOV" data set while paper [9] only uses the documents of text/html format.

We also study the performance under different parameter settings. The results are in Figure 11. In Figure 11(a), the average precision is plotted along with different α values. DirichletRank achieves best results when $\alpha = 0.96$ and $\mu = 20$. The PageRank achieves the



(a) Combination parameter α



(b) Parameters μ for DirichletRank and λ for PageRank

Figure 11: Performance comparison along with different parameters

best result when $\alpha = 0.98$ and $\lambda = 0.1$. Figure 11(b) shows the performance under different λ and μ. When μ goes to infinity or λ goes to 1, the link-based ranking will be uniform thus the performance is equal to the content-based method. Note that these empirical result show that a small λ value is preferred to ensure the effectiveness of PageRank; unfortunately, it makes PageRank more sensitive to the link farm spams. While DirichletRank is both robust against link spamming and accurate in ranking when we set $\mu = 20$.

6 Conclusions and Future Work

Link spams are created by malicious web users to beguile unbiased link analysis algorithms. Combating such spams is a major challenge for all search engines.

In this paper, we show that the popular link-based ranking algorithm PageRank has a "zero-one gap" flaw, which can be potentially exploited for spamming the results. This "zero-one gap" problem is caused by the current ad hoc way of computing the transition probabilities in the random surfing model. We propose a novel *DirichletRank* algorithm with a more principled way of computing these probabilities based on Bayesian estimation with a Dirichlet prior.

We show both analytically and empirically that DirichletRank is much more robust against

link spamming than PageRank. We evaluated the influence of bogus-page-based spams and collusion spams and compared the stability of DirichletRank and PageRank under perturbation. In all the experiments, DirichletRank is shown to be substantially more resistant to link spamming.

Moreover, experiment results also show that DirichletRank is more effective than PageRank due to its more reasonable allocation of transition probabilities. Since DirichletRank can be computed as efficiently as PageRank, it is scalable to large-scale web applications.

In the future, we plan to further extend our DirichletRank work in two directions.

First, we will develop a more general anti-spam framework. The main point for any potential anti-link-spamming algorithm is to evaluate the reliability of a hyperlink between two web pages. Assume each page p has an indicator $Q(p)$ for measuring the quality of its out-links. The higher the $Q(p)$ is, the more reliable the p's out-links are. We can further rewrite DirichletRank as:

$$\tilde{M} = diag\{1-\omega_1, ..., 1-\omega_N\} \cdot M + diag\{\omega_1, ..., \omega_N\} \cdot U$$

$$\text{where } \omega_p = \frac{\mu}{Q(p)+\mu}$$

This is a more general framework, which basically can incorporate any reasonable $Q(p)$ function. In this paper, we only discussed the simplest one, though; we plan to evaluate other more sophisticated ones in the future.

Second, we will compare our algorithm with more different variants of PageRank algorithms, such as topic-sensitive PageRank [17] and HostRank [26].

7 Acknowledgments

We thank Xin He and Gui-Rong Xue for valuable discussions and suggestions for this paper.

References

[1] AIRWeb. http://airweb.cse.lehigh.edu/.

[2] Pr0-Google's PageRank 0. http://pr.efactory.de/e-pr0.shtml.

[3] TREC. http://trec.nist.gov/.

[4] TREC-2003 Web Track. http://trec.nist.gov/data/t12.web.html.

[5] A. Benczur, K. Csalogany, T. Sallos, and M. Uher. Spamrank-fully automatic link spam detection. In *First International Workshop on Adversarial Information Retrieval on the Web (AirWeb05)*, 2005.

[6] K. Bharat and M. R. Henzinger. Improved algorithms for topic distillation in a hyperlinked environment. In *SIGIR*, pages 104–111, 1998.

[7] M. Bianchini, M. Gori, and F. Scarselli. Inside pagerank. *ACM Transaction of Internet Technology.*, 5(1):92–128, 2005.

[8] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks*, 30(1-7):107–117, 1998.

[9] D. Cai, X. He, J.-R. Wen, and W.-Y. Ma. Block-level link analysis. In *SIGIR*, pages 440–447, 2004.

[10] D. Cohn and H. Chang. Learning to probabilistically identify authoritative documents. In *ICML*, pages 167–174, 2000.

[11] T. M. Cover and J. A. Thomas. *Elements of Information Theory.* John Wiley & Sons, Inc., New York., 1991.

[12] C. H. Q. Ding, X. He, P. Husbands, H. Zha, and H. D. Simon. Pagerank: HITS and a unified framework for link analysis. Technical report, LBNL, 2002.

[13] D. Fetterly, M. Manasse, and M. Najork. Spam, damn spam, and statistics: Using statistical analysis to locate spam web pages. In *WebDB*, pages 1–6, 2004.

[14] Z. Gyongyi and H. Garcia-Molina. Link spam alliances. In *VLDB*, pages 517–528, 2005.

[15] Z. Gyongyi and H. Garcia-Molina. Web spam taxonomy. In *First International Workshop on Adversarial Information Retrieval on the Web*, 2005.

[16] Z. Gyongyi, H. Garcia-Molina, and J. Pedersen. Combating web spam with trustrank. In *VLDB*, pages 576–587, 2004.

[17] T. H. Haveliwala. Topic-sensitive pagerank. In *WWW*, pages 517–526, 2002.

[18] M. R. Henzinger, R. Motwani, and C. Silverstein. Challenges in web search engines. In *IJCAI*, pages 1573–1579, 2003.

[19] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5):604–632, 1999.

[20] L. Li, Y. Shang, and W. Zhang. Improvement of HITS-based algorithms on web documents. In *WWW*, pages 527–535, 2002.

[21] R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge University Press, United Kingdom, 1995.

[22] A. Y. Ng, A. X. Zheng, and M. I. Jordan. Stable algorithms for link analysis. In *SIGIR*, pages 258–266, 2001.

[23] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford University, November 1999.

[24] X. Wang, A. Shakery, and T. Tao. Dirichlet pagerank. In *SIGIR*, pages 661–662, 2005.

[25] B. Wu and B. D. Davison. Identifying link farm spam pages. In *WWW (Special interest tracks and posters)*, pages 820–829, 2005.

[26] G.-R. Xue, Q. Yang, H.-J. Zeng, Y. Yu, and Z. Chen. Exploiting the hierarchical structure for link analysis. In *SIGIR*, pages 186–193, 2005.

[27] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *SIGIR*, pages 334–342, 2001.

[28] C. Zhai and J. Lafferty. Two-stage language models for information retrieval. In *Proceedings of SIGIR'02*, pages 49–56, Aug 2002.

[29] H. Zhang, A. Goel, R. Govindan, K. Mason, and B. V. Roy. Impoving eigenvector-based reputation systems against collusions. In *Workshop on Algorithms and Models for the Web Graph (WAW)*, 2004.

A PROOFS

Proof of Theorem 1. Equation 3 can be proved easily from the PageRank definition, and Equation 4's proof is in [14]. Here we prove $r_s(T) \geq \frac{1}{2\lambda-\lambda^2} r_o(T)$.

$$\begin{aligned} r_s(T) &= \frac{1}{2\lambda-\lambda^2}\left[\sigma + \frac{\tau(k(1-\lambda)+1)}{N}\right] \\ &\geq \frac{1}{2\lambda-\lambda^2}\left[\sigma + \frac{\tau(2-\lambda)}{N}\right] \\ &\geq \frac{1}{2\lambda-\lambda^2}\left[\sigma + \frac{\tau}{N}\right] \\ &= \frac{1}{2\lambda-\lambda^2} r_o(T). \end{aligned}$$

Proof of Theorem 2. Since Equation 5 is the straightforward consequence of DirichletRank definition, we only show the proof of Equation 6.

According to the definition of DirichletRank, we have

$$d_s(T) = \sigma + \frac{1}{1+\mu}\sum_{\text{all B's}} d_s(B) + \frac{\tau}{N} \tag{8}$$

$$d_s(B) = \frac{1}{k+\mu} d_s(T) + \frac{\tau}{N}, \text{for all bogus pages.} \tag{9}$$

Replacing $d_s(B)$ in (8) yields

$$d_s(T) = \sigma + \frac{k}{(1+\mu)(k+\mu)} d_s(T) + \left[1 + \frac{k}{\mu+1}\right]\frac{\tau}{N}$$

then we have

$$d_s(T) = \left[1 + \frac{k}{\mu^2+(k+1)\mu}\right]\left[\sigma + \frac{k+\mu+1}{\mu+1}\frac{\tau}{N}\right]$$

Since $\frac{k}{\mu^2+(k+1)\mu} > 0$ and $\frac{k+\mu+1}{\mu+1} > 1$, we have
$r_s(T) > r_o(T)$. Hence the proof.

Proof of Theorem 3. We assume the ith bogus page has $n_i - 1$ out-links to other bogus pages and a link back to the target page. According to DirichletRank's definition,

$$d(T) = \sigma + \sum_{i=1}^{k} \frac{d(B_i)}{n_i + \mu} + \frac{\tau}{N} \tag{10}$$

Each bogus page B_i has the score

$$d(B_i) = \frac{d(T)}{k+\mu} + \sum_{j:j\to i} \frac{1}{n_j+\mu} d(B_j) + \frac{\tau}{N}$$

where $j \to i$ denotes that B_j has a link to B_i.

Summing all $d(B_i)$ together yields:

$$\begin{aligned}
& \sum_{i=1}^{k} d(B_i) = \frac{k}{k+\mu} d(T) + \sum_{i=1}^{k} \frac{n_i - 1}{n_i+\mu} d(B_i) + k\frac{\tau}{N} \\
\Rightarrow \quad & \sum_{i=1}^{k} (1 - \frac{n_i - 1}{n_i+\mu}) d(B_i) = \frac{k}{k+\mu} d(T) + k\frac{\tau}{N} \\
\Rightarrow \quad & \sum_{i=1}^{k} \frac{1+\mu}{n_i+\mu} d(B_i) = \frac{k}{k+\mu} d(T) + k\frac{\tau}{N} \\
\Rightarrow \quad & \sum_{i=1}^{k} \frac{d(B_i)}{n_i+\mu} = \frac{k}{(k+\mu)(1+\mu)} d(T) + \frac{k}{1+\mu}\frac{\tau}{N}
\end{aligned} \tag{11}$$

By replacing (11) in (10), we obtain

$$d(T) = \left[1 + \frac{k}{\mu^2 + (k+1)\mu}\right] \left[\sigma + \frac{k+\mu+1}{\mu+1}\frac{\tau}{N}\right].$$

Clearly this equation is independent of the values of n_i. Hence the proof.

Proof of Theorem 4. Assume T (B_i) has n (n_i) out-links where n (n_i) ≥ 0. Among them l (l_i) out-links point to T or other B's ($0 \leq l$ (l_i) $\leq n$ (n_i) and $l(l_i) \leq k$) and others point to outside. According to the definitions, we have:

$$d(T) = \sigma + \sum_{i:i\to T} \frac{d(B_i)}{n_i+\mu} + \frac{\tau}{N} \tag{12}$$

We use $\rightarrow$ ($\nrightarrow$) to denote there is (not) a link between two pages. For each bogus page B_i,

$$d(B_i) = \begin{cases} \frac{d(T)}{n+\mu} + \sum_{j:j\rightarrow i} \frac{d(B_j)}{n_j+\mu} + \frac{\tau}{N} & \text{if } T \rightarrow i \\ \sum_{j:j\rightarrow i} \frac{d(B_j)}{n_j+\mu} + \frac{\tau}{N} & \text{if } T \nrightarrow i \end{cases}$$

Summing all the $d(B_i)$ together yields:

$$\begin{aligned} \sum_{i=1}^{k} d(B_i) &= \frac{l}{n+\mu} d(T) + \sum_{i:i\rightarrow T} \frac{l_i - 1}{n_i + \mu} d(B_i) \\ &+ \sum_{i:i\nrightarrow T} \frac{l_i}{n_i+\mu} d(B_i) + k\frac{\tau}{N} \end{aligned} \tag{13}$$

By splitting $\sum_{i=1}^{k} d(B_i) = \sum_{i:i\rightarrow T} d(B_i) + \sum_{i:i\nrightarrow T} d(B_i)$, (13) leads to:

$$\begin{aligned} & \sum_{i:i\rightarrow T} \frac{(n_i - l_i) + (1+\mu)}{n_i+\mu} d(B_i) \\ = & \frac{l}{n+\mu} d(T) - \sum_{i:i\nrightarrow T} \frac{n_i - l_i + \mu}{n_i+\mu} d(B_i) + k\frac{\tau}{N} \\ \leq & \frac{l}{n+\mu} d(T) + k\frac{\tau}{N} \end{aligned} \tag{14}$$

$l_i \leq n_i$ results in

$$\sum_{i:i\rightarrow T} \frac{1+\mu}{n_i+\mu} d(B_i) \leq \sum_{i:i\rightarrow T} \frac{(n_i - l_i) + (1+\mu)}{n_i+\mu} d(B_i) \tag{15}$$

$l \leq n$ and $l \leq k$ result in

$$\frac{l}{n+\mu} d(T) \leq \frac{l}{l+\mu} d(T) \leq \frac{k}{k+\mu} d(T) \tag{16}$$

By combining inequity (14), (15), and (16), we obtain

$$\begin{aligned} & \sum_{i:i\rightarrow T} \frac{1+\mu}{n_i+\mu} d(B_i) \leq \frac{k}{k+\mu} d(T) + k\frac{\tau}{N} \\ \Rightarrow & \sum_{i:i\rightarrow T} \frac{d(B_i)}{n_i+\mu} \leq \frac{k \times d(T)}{(k+\mu)(1+\mu)} + \frac{k}{1+\mu}\frac{\tau}{N} \end{aligned} \tag{17}$$

Further replace the inequity (17) in the equation (12):

$$\begin{aligned} & d(T) \leq \sigma + \frac{k}{(k+\mu)(1+\mu)} d(T) + \frac{k}{1+\mu}\frac{\tau}{N} + \frac{\tau}{N} \\ \Rightarrow & \left[1 - \frac{k}{(k+\mu)(1+\mu)}\right] d(T) \leq \sigma + \frac{k+\mu+1}{1+\mu}\frac{\tau}{N} \\ \Rightarrow & d(T) \leq \left[1 + \frac{k}{\mu^2 + (k+1)\mu}\right] \left[\sigma + \frac{k+\mu+1}{\mu+1}\frac{\tau}{N}\right]. \end{aligned}$$